

**AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A method for controlling model execution in a graphical modeling environment, ~~the said~~ method comprising:

displaying a view of an executable graphical model with a plurality of executable time-based components, said executable graphical model including at least one user-configurable, executable graphical post component ~~representing an event, said graphical post component~~ having at least one input port for receiving at least one input signal, said executable graphical post component ~~specifying being configured to post an event when~~ a condition associated with ~~the said~~ at least one input signal of said executable graphical post component is satisfied;

identifying when said condition is satisfied during execution of said executable graphical model;

posting, using said executable graphical post component, said notice of an occurrence of said event in said graphical modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event; and

executing at least one executable time-based component from said plurality of executable time-based components in response to said notifying as opposed to in response to a specific point in time.

2. (Currently Amended) The method of claim 1, further comprising:

registering at least one of said plurality of executable time-based components with said event handler; and

receiving at said at least one of said plurality of executable time-based components registering with said event handler, notification of ~~the said~~ occurrence of said event following said posting.

3. (Currently Amended) The method of claim 1 wherein ~~the said~~ executable graphical post component is a block or label.

4. (Currently Amended) The method of claim 1, further comprising:

setting a sample time for an initial execution of at least one executable time-based component to be ~~the said~~ occurrence of ~~the said~~ specified event.

5. (Currently Amended) The method of claim 4, further comprising:

propagating ~~the~~said sample time to at least one other executable time-based component in said graphical model, said at least one other executable time-based component configured to inherit a sample rate.

6. (Currently Amended) The method of claim 4, further comprising:

setting a sample time of a plurality of non-contiguous executable time-based components in said graphical model to be ~~the~~said occurrence of said event.

7. (Currently Amended) The method of claim 6 wherein said sample time for ~~the~~said plurality of non-contiguous executable time-based components is set without adjusting visible connections between said executable time-based components displayed in said view.

8. (Currently Amended) The method of claim 4, further comprising:

indicating with an event ID in said view that ~~the~~said sample time of said at least one executable time-based component is set to said event.

9. (Original) The method of claim 4 wherein said event is an explicit event set by a user.

10. (Currently Amended) The method of claim 4 wherein said event is an implicit event caused by ~~the~~said execution of ~~the~~said graphical model.

11. (Currently Amended) The method of claim 10 wherein ~~the~~said implicit event is one of power-up, power-down and initialization.

12. (Currently Amended) The method of claim 10 wherein ~~the~~said implicit event corresponds to one of ~~the~~ enabling and disabling of a subsystem.

13. (Currently Amended) The method of claim 2, further comprising:

indicating which event an executable time-based component receives with a user-configurable color in said view.

14. (Currently Amended) The method of claim 1, wherein an execution scope of ~~the~~said specified event for which ~~the~~said execution of ~~the~~said graphical model is being monitored is restricted to a portion of ~~the~~said graphical model.

15. (Previously Presented) The method of claim 1 wherein each event in said graphical model maps on a one-to-one basis to an event handler, said event handler being a function.

16. (Original) The method of claim 15 wherein said function is inlined.

17. (Currently Amended) The method of claim 1 wherein a branch priority ~~block-executable time-based component~~ indicates an order of execution among at least two branches of ~~blocks~~ said executable time-based components in response to said notifying.

18. (Currently Amended) The method of claim 1 wherein more than one ~~block-executable time-based component~~ group executes in response to said notifying, said ~~block-executable time-based component~~ groups being a user selected grouping of ~~blocks~~ executable time-based components, ~~the~~said order of execution of ~~the~~ ~~block-~~said executable time-based component groups specified by a user.

19. (Currently Amended) A method for controlling model execution in a modeling environment, ~~the~~said method comprising:

displaying a view of an executable model with a plurality of executable time-based components, ~~the~~said model including at least one user-configurable, executable graphical post component ~~representing an event, the post component~~ having at least one input port for receiving at least one input signal, ~~the~~said graphical post component being configured to post an event when specifying a condition associated with ~~the~~said at least one input signal of said executable graphical post component is satisfied;

identifying when said condition is satisfied during ~~the~~said execution of said executable model;

posting, using said executable graphical post component, said notice of an occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event;

interrupting execution of an executing event in response to ~~the~~the~~said~~ determination of ~~the~~the~~said~~ occurrence of said specified event; and

performing an operation in said executable model in response to ~~the~~the~~said~~ determination of ~~the~~the~~said~~ occurrence of ~~the~~the~~said~~ specified event.

20. (Currently Amended) The method of claim 19 wherein said specified event is treated as a normal event and further comprising:

resuming execution of ~~the~~the~~said~~ interrupted event.

21. (Currently Amended) The method of claim 19 wherein said specified event is treated as an exception event and further comprising:

returning control of ~~the~~the~~said~~ execution of ~~the~~the~~said~~ model to a calling process which called ~~the~~the~~said~~ interrupted executing event without resuming execution of said interrupted event.

22. (Original) The method of claim 19 wherein said specified event is specified using an instantiated event object.

23. (Original) The method of claim 22 wherein said event is an explicit event.

24. (Original) The method of claim 22 wherein said event is an implicit event.

25. (Original) The method of claim 22 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

26. (Original) The method of claim 25 wherein said task object has at least one of a specified execution rate and priority.

27. (Currently Amended) The method of claim 26 wherein at least two events with different tasks are executing in a model and further comprising:

using event transition components to schedule ~~the said~~ execution of time-based components associated with said at least two events, said event transition components separating ~~the said~~ execution of said time-based components associated with said at least two events.

28. (Currently Amended) The method of claim 19 wherein ~~the said~~ operation is controlled by an order of execution indicated in a branch priority block.

29. (Currently Amended) The method of claim 19 wherein ~~the said~~ operation is ~~the said~~ execution of more than one ~~block-executable time-based component~~ groups, said ~~block-executable time-based component~~ groups -being a user selected grouping of ~~blocks-executable time-based components~~, ~~the said~~ order of execution of ~~the block-said executable time-based component~~ groups specified by a user.

30-32. (Canceled)

33. (Currently Amended) A physical computer-readable medium holding computer-executable instructions for controlling model execution in a graphical modeling environment, ~~the said~~ instructions comprising:

one or more instructions for displaying a view of an executable graphical model with a plurality of executable time-based components, said executable graphical model including at least one user-configurable, executable graphical post component ~~representing an event, the graphical post component~~ having at least one input port for receiving at least one input signal, ~~the said executable~~ graphical post component being configured to post an event when specifying a condition associated with ~~the said~~ at least one input signal of said executable graphical post component is satisfied;

one or more instructions for identifying when said condition is satisfied during ~~the said~~ execution of said executable graphical model;

one or more instructions for posting, using said executable graphical post component, said notice of an occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event; and

one or more instructions for executing at least one executable time-based component from said plurality of executable time-based components in response to said notifying as opposed to in response to a specific point in time.

34. (Currently Amended) The medium of claim 33, wherein ~~the~~said instructions further comprise:

one or more instructions for registering at least one of said plurality of executable time-based components with said event handler; and

one or more instructions for receiving at ~~the~~said at least one of said plurality of executable time-based components registering with said event handler notification of ~~the~~said occurrence of said event following said posting.

35. (Currently Amended) The medium of claim 33, wherein ~~the~~said executable graphical post component is a block or label.

36. (Currently Amended) The medium of claim 33, wherein ~~the~~said instructions further comprise :

one or more instructions for setting a sample time for an initial execution of at least one executable time-based component to be ~~the~~said occurrence of ~~the~~said specified event.

37. (Currently Amended) The medium of claim 36, wherein ~~the~~said instructions further comprise:

one or more instructions for propagating ~~the~~said sample time to at least one other executable time-based component in said graphical model, said at least one other executable time-based component configured to inherit a sample rate.

38. (Currently Amended) The medium of claim 36, wherein ~~the~~said instructions further comprise:

one or more instructions for setting a sample time of a plurality of non-contiguous executable time-based components in said graphical model to be ~~the~~said occurrence of said event.

39. (Currently Amended) The medium of claim 38 wherein said sample time for ~~the~~said plurality of non-contiguous executable time-based components is set without adjusting visible connections between executable time-based components displayed in said view.

40. (Currently Amended) The medium of claim 36, wherein ~~the~~said instructions further comprise:

one or more instructions for indicating with an event ID in said view that ~~the~~said sample time of said at least one executable time-based component is set to said event.

41. (Original) The medium of claim 36 wherein said event is an explicit event set by a user.

42. (Currently Amended) The medium of claim 36 wherein said event is an implicit event caused by ~~the~~said execution of ~~the~~said graphical model.

43. (Currently Amended) The medium of claim 42 wherein ~~the~~said implicit event is one of power-up, power-down and initialization.

44. (Currently Amended) The medium of claim 42 wherein ~~the~~said implicit event corresponds to one of ~~the~~said enabling and disabling of a subsystem.

45. (Currently Amended) The medium of claim 34, wherein ~~the~~said instructions further comprise:

one or more instructions for indicating which event a executable time-based component receives with a user-configurable color in said view.

46. (Currently Amended) The medium of claim 33, wherein an execution scope of ~~the~~said specified event for which ~~the~~said execution of ~~the~~said graphical model is being monitored is restricted to a portion of ~~the~~said graphical model.

47. (Previously Presented) The medium of claim 33 wherein each event in said graphical model maps on a one-to-one basis to an event handler, said event handler being a function.

48. (Original) The medium of claim 47 wherein said function is inlined.

49. (Currently Amended) The medium of claim 33 wherein a branch priority block indicates an order of execution among at least two branches of ~~blocks~~ executable time-based components in response to said notifying.

50. (Currently Amended) The medium of claim 33 wherein more than one ~~block-executable time-based component~~ groups execute in response to said notifying, said ~~block-executable time-based component~~ groups being a user selected grouping of ~~blocks~~ the executable time-based components, ~~the said~~ order of execution of ~~the block~~ said executable time-based component groups specified by a user.

51. (Currently Amended) A physical computer-readable medium holding computer-executable instructions for controlling model execution, ~~the said~~ instructions comprising:

one or more instructions for displaying a view of an executable model with a plurality of executable time-based components, ~~the said~~ model including at least one user-configurable, executable graphical post component ~~representing an event, the post component~~ having at least one input port for receiving at least one input signal, ~~the said~~ post component being configured to post an event when specifying a condition associated with ~~the said~~ at least one input signal of said executable graphical post component is satisfied;

one or more instructions for identifying ~~the said~~ satisfaction of said specified condition during ~~the said~~ execution of said executable model;

one or more instructions for posting, using said executable graphical post component, ~~the said~~ notice of an occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of ~~the said~~ occurrence of said event;

one or more instructions for interrupting execution of an executing event in response to ~~the said~~ determination of ~~the said~~ occurrence of said specified event; and

one or more instructions for performing an operation in said executable model in response to ~~the said~~ determination of ~~the said~~ occurrence of ~~the said~~ specified event.

52. (Currently Amended) The medium of claim 51 wherein said specified event is treated as a normal event and wherein ~~the said~~ instructions further comprise:



one or more instructions for resuming execution of ~~the~~said interrupted event.

53. (Currently Amended) The medium of claim 51 wherein said specified event is treated as an exception event and wherein ~~the~~said instructions further comprise:

one or more instructions for returning control of ~~the~~said execution of ~~the~~said model to a calling process which called ~~the~~said interrupted executing event without resuming execution of said interrupted event.

54. (Original) The medium of claim 51 wherein said specified event is specified using an instantiated event object.

55. (Original) The medium of claim 54 wherein said event is an explicit event.

56. (Original) The medium of claim 54 wherein said event is an implicit event.

57. (Original) The medium of claim 54 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

58. (Original) The medium of claim 57 wherein said task object has at least one of a specified execution rate and priority.

59. (Currently Amended) The medium of claim 58 wherein at least two events with different tasks are executing in a model and wherein ~~the~~said instructions further comprise:

one or more instructions for using event transition components to schedule ~~the~~said execution of executable time-based components associated with said at least two events, said event transition components separating ~~the~~said execution of said executable time-based components associated with said at least two events.

60. (Currently Amended) The medium of claim 51 wherein ~~the~~said operation is controlled by an order of execution indicated a branch priority block.

61. (Currently Amended) The medium of claim 51 wherein ~~the~~said operation is ~~the~~said execution of more than one ~~block~~executable time-based component groups, said ~~block~~executable time-based component groups- being a user selected grouping of ~~blocks~~said executable time-based components, ~~the~~said order of execution of ~~the block~~said executable time-based component groups specified by a user.